

Requested Patent: JP2002073393A

Title:

SYSTEM FOR MULTILAYER LOGICAL VOLUME MANAGING SYSTEM AND ITS METHOD ;

Abstracted Patent: JP2002073393 ;

Publication Date: 2002-03-12 ;

Inventor(s):

RAFANELLO BENEDICT MICHAEL;; PELOQUIN MARK A;; KUWON FUU TORAN;;
CHRISTY NESBITT ULMAN ;

Applicant(s): INTERNATL BUSINESS MACH CORP ;

Application Number: JP20010112905 20010411 ;

Priority Number(s): ;

IPC Classification: G06F12/00; G06F3/06; G11B27/00 ;

Equivalents: GB2373075 ;

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a system of a multi-layer logical volume management which allows an aggregation of more than one level and provide its method. SOLUTION: A mono-layer logical volume model of technique of the present time is extended by the system for a multi-layer logical volume management system (LVM) and its method to operate the aggregation of more than one level. A limit peculiar to each aggregating technique can be bypassed by using different kinds of aggregations such as drive-link, mirror-ring, software RAID and the like together according to the aggregation of more than one level. A LVM data area in which information on formation of the multi-layer logical volume is stored is housed in the last partition of a logical volume. The LVM data area is found with using a broadcasting method, and when combined with a false extended-boot record, it is possible that each level of the aggregations seems a single partition to the next upper aggregator. An unlimited number of aggregation layers, therefore, can be executed within one logical volume.

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-73393

(P2002-73393A)

(43)公開日 平成14年3月12日 (2002.3.12)

(51)Int.Cl. ⁷	識別記号	F I	特許出願公開番号
G 0 6 F 12/00	5 2 0	G 0 6 F 12/00	5 2 0 J 5 B 0 6 5
	5 0 1		5 0 1 A 5 B 0 8 2
	5 1 4		5 1 4 E 5 D 1 1 0
3/06	5 4 0	3/06	5 4 0
G 1 1 B 27/00		G 1 1 B 27/00	D
審査請求 有 請求項の数19 O L (全 13 頁)			

(21)出願番号 特願2001-112905(P2001-112905)

(22)出願日 平成13年4月11日 (2001.4.11)

(31)優先権主張番号 09/561184

(32)優先日 平成12年4月27日 (2000.4.27)

(33)優先権主張国 米国 (US)

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 ベネディクト・マイケル・ラファネロ
アメリカ合衆国78681 テキサス州ラウン
ド・ロック ウィット・コーヴ 4012

(74)代理人 100086243

弁理士 坂口 博 (外2名)

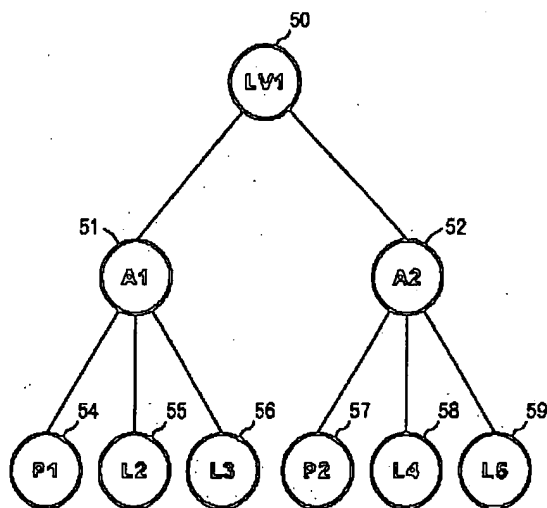
最終頁に続く

(54)【発明の名称】 複数層論理ボリューム管理システムのシステムおよび方法

(57)【要約】

【課題】 複数レベルのアグリゲーションを許容する複数層論理ボリューム管理のシステムおよび方法を提供すること。

【解決手段】 複数層論理ボリューム管理 (「LVM」) システムのシステムおよび方法によって、複数レベルのアグリゲーションを扱うために現在の技術の単一層論理ボリューム・モデルを拡張する。複数レベルのアグリゲーションによって、ドライブ・リンク、ミラーリング、およびソフトウェアRAIDなどの複数の種類のアグリゲーションを一緒に使用して、各個々のアグリゲーション技術に固有の制限を迂回することができる。複数層論理ボリュームの編成に関する情報が格納されるLVMデータ域が、論理ボリュームの最後のパーティション内に格納される。ブロードキャスト方法を使用して、LVMデータ域を突きとめ、偽拡張ブート・レコードと結合された時に、アグリゲーションの各レベルが、次の上位アグリゲータに単一パーティションに見えるようにすることができる。したがって、無限の個数のアグリゲーション層を、1つの論理ボリューム内で実施することができる。



【特許請求の範囲】

【請求項1】コンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法であって、

複数のコンピュータ可読媒体パーティションを設けるステップと、

前記複数のコンピュータ可読媒体パーティションの複数のパーティション・アグリゲーションへの第1アグリゲーションを実行するステップと、

機能強化された可用性およびコンピュータ・システムへのアクセス可能性のために、前記コンピュータ可読パーティションが、単一のコンピュータ可読エンティティにグループ化されるように、前記パーティション・アグリゲーションを複数層論理ボリュームにアグリゲートするステップとを含む方法。

【請求項2】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、前記媒体パーティション間のドライブ・リンクを設けるステップを含む、請求項1に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

【請求項3】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、少なくとも1つのRAIDアレイを設けるステップを含む、請求項1に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

【請求項4】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、前記媒体パーティション間のディスク・ミラーリングを設けるステップを含む、請求項1に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

【請求項5】前記複数層論理ボリュームの編成を示す情報を含む論理ボリューム・データ域を設けるステップと、

前記コンピュータ可読媒体パーティションの1つの中で拡張ブート・レコード内のリンクを設けるステップであって、前記リンクが、前記論理ボリューム内の前記論理ボリューム・データ域の位置を示すステップと、

前記拡張ブート・レコードから、前記リンクを取り出すステップと、前記複数層論理ボリュームの前記編成を判定できるように、前記論理ボリューム・データ域から前記情報を取り出すステップとをさらに含む、請求項1に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

【請求項6】前記論理ボリューム・データ域から前記情報を取り出す前記ステップが、さらに、少なくとも1つのアグリゲータに要求をブロードキャストすることを含む、請求項5に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する

方法。

【請求項7】前記論理ボリューム・データ域を前記論理ボリュームの最後のパーティションの末尾に格納するステップをさらに含む、請求項5に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

【請求項8】コンピュータ可読媒体の論理ボリュームを作成し、管理するシステムであって、

少なくとも1つのコンピュータ可読媒体パーティションを有する少なくとも1つのコンピュータ可読媒体装置と、

コンピュータ・ソフトウェアを実行することができ、少なくとも1つのコンピュータ可読媒体装置にインターフェースされる、コンピュータ・プロセッサと、

前記コンピュータ可読媒体パーティションの複数層のアグリゲーションが、前記コンピュータ・プロセッサおよび前記コンピュータ・プロセッサによって実行されるソフトウェアによって、単一のエンティティとして論理的にアクセス可能になるように組み合わせられるように、複数層のコンピュータ可読媒体パーティション・アグリゲータを含む複数層論理ボリューム・マネージャとを含むシステム。

【請求項9】前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、コンピュータ可読媒体のRedundant Array of Independent Disks (「RAID」)アレイを含む、請求項8に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

【請求項10】前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、ディスク・ミラーリング・サブシステムを含む、請求項8に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

【請求項11】前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、ドライブ・リンク・サブシステムを含む、請求項8に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

【請求項12】前記複数層論理ボリューム・マネージャが、機能強化されたIBM OS/2 LVMを含む、請求項8に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

【請求項13】コンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラムであって、前記論理ボリュームが、複数のコンピュータ可読媒体パーティションを有し、前記コンピュータ・プログラムが、前記複数のコンピュータ可読媒体パーティションの複数のパーティション・アグリゲーションへの第1アグリゲーションを実行する、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、

機能強化された可用性およびコンピュータ・システムへ

のアクセス可能性のために、前記コンピュータ可読パーティションが、単一のコンピュータ可読エンティティにグループ化されるように、前記パーティション・アグリゲーションを複数層論理ボリュームにアグリゲートする、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体とを含むコンピュータ・プログラム。

【請求項14】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、前記媒体パーティション間のドライブ・リンクを設けるコンピュータ可読プログラム・コード手段をさらに含む、請求項13に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【請求項15】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、少なくとも1つのRAIDアレイを設けるコンピュータ可読プログラム・コード手段をさらに含む、請求項13に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【請求項16】コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、前記媒体パーティション間のディスク・ミラーリングを設けるコンピュータ可読プログラム・コード手段をさらに含む、請求項13に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【請求項17】前記複数層論理ボリュームの編成を示す情報を含む論理ボリューム・データ域を設ける、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、

前記コンピュータ可読媒体パーティションの1つの中で拡張ブート・レコード内のリンクを設けるコンピュータ可読プログラム・コード手段であって、前記リンクが、前記論理ボリューム内の前記論理ボリューム・データ域の位置を示す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、

前記拡張ブート・レコードから、前記リンクを取り出す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、前記複数層論理ボリュームの前記編成を判定できるように、前記論理ボリューム・データ域から前記情報を取り出す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体とをさらに含む、請求項13に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成

し、管理するコンピュータ・プログラム。

【請求項18】前記論理ボリューム・データ域から前記情報を取り出す前記コンピュータ可読プログラム・コード手段が、さらに、少なくとも1つのアグリゲータに要求をブロードキャストするプログラム・コード手段を含む、請求項17に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【請求項19】前記論理ボリューム・データ域を前記論理ボリュームの最後のパーティションの末尾に格納するコンピュータ可読プログラム・コード手段をさらに含む、請求項17に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・ディスク媒体、コンピュータ・ディスクのフォーマット、オペレーティング・システムおよびデバイス・ドライバによるコンピュータ可読媒体の編成、およびコンピュータ・ディスクの論理ボリュームの管理の分野に関する。

【0002】

【従来の技術】コンピュータ・システムの永続記憶装置および大容量記憶装置、特にパーソナル・コンピュータに使用される装置は、当技術分野で周知である。多数が、フロッピー・ディスク、取外し可能ハード・ディスク・ドライブ（「HDD」）、およびコンパクトディスク読取専用メモリ（「CD-ROM」）などのディスクベースである。図1に、通常のパーソナル・コンピュータ・システム1のアーキテクチャを示す。ここで、CPU2が、キーボード3、モニタまたはディスプレイ5、およびマウス4などのさまざまな入出力装置とインターフェースする。CPU2は、CD-ROMドライブ7、ハード・ディスク・ドライブ6、およびフロッピー・ドライブ5を含む複数の記憶周辺機器とインターフェースすることもできる。通常、フロッピー・ディスク・ドライブは、Integrated Drive Electronics（「IDE」）8を介してCPUとインターフェースするが、このインターフェースは、その代わりに、複数の他の標準インターフェースまたは独自インターフェースの1つとすることができる。ハード・ディスク・ドライブ6およびCD-ROMドライブ7は、図示のように（9）、IDEまたはSmall Computer System Interface（「SCSI」）を介してCPU2とインターフェースすることができる。

【0003】図2に、パーソナル・コンピュータ・システム20のハードウェア、ファームウェア、およびソフトウェアの編成の一般化を示す。ハードウェア・グループ21には、上で述べた永続記憶装置、ならびに、リアルタイム・クロック、キーボード・コントローラ、ディスプレイ・アダプタなどの他のシステム・ハードウェア

構成装置が含まれる。基本入出力システム(「BIOS」)22が、通常はこれらのシステム構成装置の直接ファームウェア制御を提供する。IBM OS/2オペレーティング・システムなどのオペレーティング・システム24が、マルチタスキングまたはマルチスレッド式スケジューリングとシステム・アプリケーション・プログラム25の優先順位付けを含むシステム・リソースの高水準管理を提供する。ドライバ23が、メーカー固有およびモデル固有のLANインターフェース・カード・ドライバまたはCD-Rewritable(「CD-RW」)ドライバなどの、特定のハードウェア用の特定の高标准インターフェース機能および制御機能を提供する。このシステムの一般化された視点は、Microsoft Windows、UNIX(登録商標)、またはLinuxなどのさまざまなオペレーティング・システムを使用する、ワークステーションなどの代替の非IBM互換プラットフォームのシステムにも適用される。このコンピュータ・システム・リソースおよびソフトウェア機能性の全般的な編成は、当技術分野で明確に理解されている。

【0004】図3に移ると、ハード・ディスク・ドライブ、フロッピー・ディスク、およびCD-ROMなどのディスクベースの大容量記憶装置は、物理的には、回転する記憶トラック30に基づく。このトラックは、フロッピー・ディスクなどの柔軟なマイラからなるか、またはハード・ディスク・ドライブおよびCD-ROMなどのア

ルミニウム、ガラス、またはプラスチックからなるより固いプラッタとすることができる。磁気媒体の場合、プラッタの一面または両面に、読書ヘッドからの磁気パルスを記録することができる磁性体層がコーティングされる。光媒体の場合、データ記録は、光の帯域の反射率の変化を使用して行われ、この変化が、レーザベースのヘッドによって読み取られる。書込可能および書換可能なCD-ROMドライブでは、磁気ディスクの技術と光ディスクの技術が組み合わされる。しかし、一般に、ディスク上のデータの編成は、類似する。ディスク表面が、複数の同心円またはトラック31に分割される。ハード・ディスク・ドライブなどの一部のディスク・ドライブは、複数のプラッタからなり、この場合には、各プラッタの対応するトラックが、シリンドラにグループ化される。各トラックは、複数のセクタ32に分割され、このセクタにデータを格納することができる。

【0005】図4に移ると、0から「n」までの番号を付けられた順序付けられたセクタの集合として表されたコンピュータ・ディスク・ドライブ41が示されている。ハード・ドライブの最初のセクタであるセクタ0に、マスタ・ブート・レコード(「MBR」)が含まれる。MBRには、ディスクの残りのパーティション定義が含まれる。表1に、例の部分的なMBRを示す。

【0006】

【表1】

表1 8GBドライブのパーティション・テーブル			
パーティション	開始 (シリンドラ、サイド、セクタ)	終了 (シリンドラ、サイド、セクタ)	長さ (セクタ)
第1	0, 1, 1	391, 254, 63	6297417
第2	392, 0, 1	783, 254, 63	6297480

【0007】表1に示されたディスク分割化では、MBRが、サイド0、シリンドラ0、セクタ1にある最初のセクタに配置される。MBRは、1セクタだけを必要とするが、63セクタのトラック全体が、MBRの使用のために「ブロック」され、サイド0、シリンドラ0の62セクタが、未使用のまま残される。

【0008】パーティション・テーブルは、その中に、2種類のパーティションすなわち、基本パーティションと拡張パーティションを定義する項目を有する。従来のディスク・フォーマット方式では、1つの拡張パーティション411だけを定義することができる。P1 43およびP2 44は、基本パーティションである。基本パーティションおよび拡張パーティションの順序および位置は、変更することができるが、MBRのパーティション・テーブル内に、それらを定義する項目が必ず存在する。

【0009】拡張パーティション411は、MBRパーティション・テーブル内の単一の項目を使用して、単一のパーティションとしてMBRのパーティション・テ

ブル内で定義される。基本的に、MBR内のこの項目は、この拡張パーティションの内部で、他のパーティションおよびパーティション定義を見つけることができることをコンピュータ・オペレーティング・システムに示すだけである。オペレーティング・システムは、通常、これらのパーティションまたはパーティションのグループに、論理ドライブ名または論理ボリュームもしくはその両方を割り当てる。

【0010】拡張パーティション内のパーティションのサイズおよび位置を判定するために、オペレーティング・システムは、拡張パーティションの最初のセクタにアクセスするが、このセクタには、通常は、拡張ブート・レコード(「EBR」)と称するもう1つのブート・レコードが含まれる。EBRのフォーマットは、MBRのフォーマットに類似し、やはり当技術分野で周知である。

【0011】図4に、拡張パーティション411内の第1EBR45、第2EBR47、および第3EBR49を示す。実際には、1つの拡張パーティション内に、こ

れより多数または少数のEBRがある場合がある。

【0012】各EBRには、MBRパーティション・テーブルに類似するパーティション・テーブルが含まれる。従来、パーソナル・コンピュータおよびワークステーションで一般的に使用されるコンピュータ・ドライブについて、各EBR内で2つの項目だけが使用中になる。一方の項目では、論理パーティションが定義され、第2の項目は、次のEBRへのリンクまたはポインタとして働く。図4に、第1EBR45の第2項目から第2EBR47の先頭へのポインタ412と、第2EBR47の第2項目から第3EBR49の先頭への同様のポインタ413を示す。拡張パーティションの最後のEBRには、後続のEBRへのポインタが含まれず、これによって、それが拡張パーティション内の最後のEBRであることがオペレーティング・システムに示される。この形で、オペレーティング・システムは、決定的な形で拡張パーティション内の無制限の数のパーティションまたは論理ドライブの定義を見つけ、突きとめることができる。

【0013】各パーティション・テーブル項目には、それがEBRであれMBRであれ、オペレーティング・システムに、そのディスクで使用されるフォーマットまたはファイル・システムを示すフィールドがある。たとえば、DOS (「disk operating system」) システムの場合、このフィールドによって、ファイル・システムがFile Allocation Table (「FAT」) フォーマットであることを示すことができる。また、IBM社のOS/2オペレーティング・システムが動作するシステムの場合、この項目によって、ファイル・システムがハイパフォーマンス・ファイル・システム (「HPFS」) フォーマットであることを示すことができる。当業界には、通常は、Microsoft社のWindows、IBM社のOS/2およびAIX、UNIXの変形、およびLINUXなど、コンピュータ用の一般的なオペレーティング・システムに関連する、複数の周知のファイル・システム・フォーマットがある。このフィールドを使用して、オペレーティング・システムは、コンピュータ・ディスク上の基本パーティションおよび拡張パーティションのパーティション内に格納されたデータ・ファイルを見つけ、アクセスする方法を判定することができる。このファイル・システム・フォーマット・インジケータを、「システム・インジケータ」と称する場合がある。

【0014】IBM社のOS/2オペレーティング・システムには、論理ボリューム・マネージャまたは「LVM」と称する機能が含まれる。LVMがないシステムの場合、オペレーティング・システムによって使用可能なパーティションのそれぞれに、「C:」または「F:」などのドライブ名が割り当てられ、コンピュータ・システム内のディスク上の各パーティションに関連するドライブ名が作られる。この名前を割り当てる処理は、一般

に既知である。LVMを有するシステムでは、その代わりに、ドライブ名を、1つまたは複数のパーティションを含めることができる論理ボリュームにマッピングすることができる。この、パーティションを単一のエンティティに組み合わせる処理を、包括的に「アグリゲーション」と称する。OS/2 LVMの高度なモジュラー設計によって、アグリゲーションを実行する機能性が、LVMプログラムの単一のモジュール内に完全に含まれる。LVMでは、アグリゲーションを実行するモジュールのすべてを、「アグリゲータ」と呼ぶ。

【0015】アグリゲーションには、ドライブ・リンク、ミラーリング、ソフトウェアRedundant Array of Independent Disks (「RAID」) など、さまざまな形態がある。OS/2 LVMでは、ドライブ・リンクの使用を介する単一レベルのアグリゲーションが可能である。内部的には、OS/2 LVMは、階層化モデルを使用する。ボリュームに対する使用のためにLVMによって提供される機能のそれぞれが、LVM内の1層である。層への入力、層からの出力と同一の形態および構造を有する。ボリュームに対して使用される層は、スタックを形成し、入出力要求は、最上位層からスタックを下って最下位層まで処理される。現在、最下位層は、パス・スルー層と称する特殊な層である。最上位層は、必ずアグリゲータであり、現在の実施形態では、必ずドライブ・リンク層になる。スタックの途中のすべての層が、不良ブロック再配置などの非アグリゲーション機能を表す。

【0016】図9に、LVMの階層化モデルと、物理パーティションの論理ボリューム90へのアグリゲーションの関係を示す。左側に、「機能スタック」が示されており、これは、ディスク・デバイスまたはデバイス・ドライバに直接にインターフェースする「パス・スルー」層97を最下部に有する。パス・スルー層97の上に、不良ブロック再配置 (「BBR」) などの機能層96を設けることができる。この機能の上に、ドライブ・リンクなどのアグリゲーション層95を設けることができる。機能スタック・モデルの視点からは、入出力要求98は、スタックの最上部で受け取られ、下に向かってパス・スルー層まで伝搬される。それを論理ボリューム90のツリー・モデルと比較すると、アグリゲータA191は、アグリゲーション層95に対応し、機能層96は、アグリゲータA191とそのパーティション定義P192、P293、およびP394の間の3つのインターフェースに対応し、パス・スルー層97は、パーティション定義と実際のデバイスまたはデバイス・ドライバの間のインターフェースに対応する。この種のLVM構造、機能スタック・モデル、およびツリー・モデルは、当技術分野で十分に理解され、このモデルは、Hewlett Packard社のHP-UXおよびIBM社のAIXなどの他のオペレーティング・システム内の論理ボリ

ューム管理システムにも同等によく適用することができる。

【0017】論理ボリュームの一部であるパーティションは、特殊なファイルシステム・フォーマット・インジケータを有する。このインジケータは、既存のどのファイルシステムにも対応せず、論理ボリュームに属するものとしてそのパーティションを識別するように働く。論理ボリュームの実際のファイルシステム・フォーマット・インジケータは、別の場所に格納される。さらに、ボリュームに属するパーティションは、ボリューム内の各パーティションの末尾に、LVMデータ域を有する。LVMデータ域に格納されたデータを用いて、LVMが、システムがブートされるたびにボリュームを再作成できるようにする。したがって、LVMを用いると、パーティションのグループ化を、オペレーティング・システムにとって、単一のドライブ名割当を有する単一のエンティティとして見えるようにすることができる。

【0018】OS/2オペレーティング・システムの以前のバージョンでは、FORMATディスク・ユーティリティなどのファイル・システム・ユーティリティが、低水準入出力制御（「IOCTL」）機能を介してフォーマットされるパーティションのパーティション・テーブルにアクセスした。システムは、IOCTLを提供して、ソフトウェア・アプリケーションが、ファイルベースの操作を使用するのではなく、ファイル・システムを迂回してコンピュータ・ディスクを直接読み書きできるようにする。IOCTL機能を使用して、アプリケーション・プログラムは、実際に、処理されるパーティションを定義するEBRからパーティション自体の末尾までのすべてのものにアクセスすることができる。これによって、ディスク・ユーティリティが、処理中のパーティションに対応するパーティション・テーブル項目を見つけることができ、それを変更できるようになる。たとえば、FORMATは、それが成功裡にフォーマットする各パーティションのパーティション・テーブル項目のファイルシステム・フォーマット・インジケータを更新する。この方法は、個々のパーティションの処理については良好に動作するが、論理ボリュームを扱う時には問題が生じる。論理ボリュームは、システムにとって単一のエンティティに見えるが、これは、論理ボリュームが、以前のディスク・ユーティリティにはパーティションのように見え、それらのユーティリティが、当然、論理ボリュームをパーティションとして扱おうとすることを意味する。しかし論理ボリュームには複数のパーティションが含まれる場合があるので、それを記述したEBRまたはパーティション・テーブル項目が存在しない。以前のディスク・ユーティリティが、論理ボリュームに含まれるパーティションの1つのEBRまたはパーティション・テーブル項目へのアクセスを許可される場合には、そのパーティション・テーブル項目に記述されたパーティションが、

ディスク・ユーティリティがパーティションとみなすものと一致しなくなる。さらに、FORMATがファイルシステム・フォーマット・インジケータを更新する時など、ディスク・ユーティリティが、パーティション・テーブル項目を変更する場合に、結果のパーティション・テーブル項目が、正しくなくなる。したがって、以前のディスク・ユーティリティが、論理ボリュームに含まれるパーティションのEBRまたはパーティション・テーブル項目にアクセスすることを許可されてはならないが、以前のディスク・ユーティリティは、正しく機能するためにEBRおよびパーティション・テーブル項目を必要とする。

【0019】OS/2 LVMの最初のバージョンでは、この問題が、論理ボリューム全体を単一の区画であるかのように記述する「偽」パーティション・テーブル項目を含む「偽」EBRを作成することによって解決された。この「偽」EBRは、論理ボリュームの内部で、論理ボリューム内の最初のパーティションに格納された。IOCTL機能が、インターセプトされ、EBRに関するすべての要求が、「偽」EBRにリダイレクトされた。これによって、論理ボリュームを、以前のディスク・ユーティリティによってパーティションとして扱うことができ、これによって、以前のディスク・ユーティリティが機能できるようになった。

【0020】現在使用可能なOS/2 LVMの設計では、単一層のアグリゲーションだけがサポートされる。これによって、行うことができるものに制限が課される。たとえば、アグリゲータとしてソフトウェアRAIDを使用する場合、単一のボリュームにアグリゲートすることができるパーティションの数に制限が生じる。しかし、複数レベルのアグリゲーションが許容される場合には、ドライブ・リンクを使用して、複数のソフトウェアRAIDアグリゲートを1つのボリュームにアグリゲートすることができ、これによって、ソフトウェアRAIDの制限なしに、ソフトウェアRAIDのすべての長所を有するボリュームが提供されるはずである。

【0021】

【発明が解決しようとする課題】したがって、当技術分野には、複数レベルのアグリゲーションを許容する複数層論理ボリューム管理のシステムおよび方法の必要が存在する。さらに、当技術分野には、IOCTL機能などの既存のディスク・ユーティリティ機能との互換性を有する、複数層論理ボリューム管理のシステムおよび方法の必要が存在する。

【0022】

【課題を解決するための手段】本発明の前述および他の目的、特徴、および長所は、添付図面に図示された、本発明の好ましい実施形態の以下の詳細な説明から明らかになる。添付図面では、同様の符号が、本発明の同様の部分を表す。

【0023】複数層論理ボリューム管理システムのシステムおよび方法は、OS/2 LVMおよび類似システムによって使用される既存の論理ボリューム管理モデルを拡張して、複数レベルのアグリゲーションを処理できるようにする方法を提供する。複数レベルのアグリゲーションによって、ドライブ・リンク、ミラーリング、およびソフトウェアRAIDなどの複数のアグリゲータと一緒に使用して、各個々のアグリゲーション技術に固有の制限を迂回することが可能になる。一例として、多数のソフトウェアRAID実施形態は、単一のエンティティに組み合わせることができるパーティションの数に対する制限を有する。しかし、ドライブ・リンクを使用して複数のソフトウェアRAIDエンティティを単一のボリュームに組み合わせることによって、ボリュームが、ソフトウェアRAIDの長所を有すると同時に、ソフトウェアRAID自体が許容する数より多数のパーティションを使用することができる。

【0024】

【発明の実施の形態】本発明は、IBM社のOS/2オペレーティング・システムが稼動する、IBMパーソナル・コンピュータなどの周知のコンピューティング・プラットフォームを使用して実現されることが好ましい。しかし、本発明は、本発明の趣旨および範囲から逸脱せずに、Microsoft Windows、HP-UX、UNIX、またはLinuxなどの代替オペレーティング・システムが稼動する、Sun Microsystems社のワークステーションまたはIBM RS/6000ワークステーションなどの他の一般的なコンピュータ・システム・プラットフォームで実現することができる。

【0025】好ましい実施形態のオペレーティング・システムとしてOS/2を使用することによって、前に「発明の属する技術分野」で説明した既存のOS/2論理ボリューム・マネージャ（「LVM」）を変更して、本発明を実現することができる。既存のLVMは、ドライブ・リンク層と称する単一層のアグリゲーションと、論理ボリュームを作成し管理するシステムを提供する。既存のLVMは、階層化モデルを使用し、ボリュームに対して使用するために使用可能な特徴または機能のそれぞれが、別々のモジュールであり、これらのモジュールの入力および出力が、形態および構造において同一である。ボリュームに適用される機能は、スタックを形成し、アグリゲータ（ドライブ・リンク）が、スタックの最上位層であり、特殊なバス・スルー層が、スタックの最下位層である。ボリュームが作成される（または、システム・ブート後に再作成される）時に、機能スタックが、バス・スルー層から始めて最下位から上へと作成される。ボリュームのすべてが作成され、使用の準備ができた時に、LVMが、システム内のボリュームに対する入出力要求の処理を開始しなければならない。入出力要求が処理される時には、その入出力要求は、スタックの

最上部から層を介して下へ（ある層の出力が次の層の入力になる）、機能スタックの最下部へと処理され、最下部からデバイスに送られる。

【0026】複数層論理ボリュームの作成および構造ボリュームが作成される（またはリブートの後に再作成される）時に、そのボリューム内の各パーティション末尾にあるLVMデータ域が、調べられる。LVMデータ域には、ボリューム上で使用中の機能、ボリュームの機能スタックにその機能が現れる順序、および各機能の初期化に必要なデータの位置（LVMデータ域内の）をリストした固定サイズのテーブルがある。LVMは、このテーブルを使用して、そのボリュームの機能スタックを作成する。各機能が機能スタックに追加される際に、その機能は、それが操作することになるパーティションのLVMデータ域からのデータを使用して初期化される。最後に追加される機能は、アグリゲータすなわちドライブ・リンクである。ドライブ・リンクは、LVMデータ域からのデータを使用して初期化され、ボリュームを構成するパーティションからなる単一のアグリゲータを作る。ドライブ・リンクによって作られるアグリゲータは、LVMデータ域を関連付けられないことを除いて、パーティションであるように見え、そのサイズは、アグリゲータ内のパーティションのサイズの合計にほぼ等しい。

【0027】複数レベルのアグリゲーションを処理するようにこのモデルを拡張するために、本発明は、アグリゲータがパーティションのように見え、その結果、各アグリゲータがパーティションをアグリゲートしていると考えようにすることを要求する。これは、各アグリゲータにLVMデータ域を追加し、その後、このLVMデータ域を、パーティションに常駐しているかのように扱うことを必要とする。現在、パーティションに対する機能を初期化するのに使用されるデータは、そのパーティションのLVMデータ域に格納される。アグリゲーションが1レベルだけであり、機能スタックの最上位機能がアグリゲータに制限される状態では、ボリュームに対する機能のすべてを初期化するのに必要なデータのすべてが、そのボリューム内のパーティションのそれぞれのLVMデータ域に存在する。アグリゲータがパーティションのように見えるようになるので、アグリゲータを操作する機能を有することが可能になる。アグリゲータに対するこれらの機能を初期化するのに必要なデータは、そのアグリゲータのLVMデータ域にあることになる。アグリゲータは、パーティションのように見えるので、機能の初期化処理は、それがパーティションとアグリゲータのどちらに対して使用される場合でも同一になる。機能をアグリゲータに適用することができ、アグリゲータが機能であるから、アグリゲータを、他のアグリゲータによって作成されるアグリゲータに適用することができる。したがって、本発明によって、無制限のレベル数の

アグリゲーションが可能になる。

【0028】たとえば、図5の論理ボリュームには、2つのアグリゲーション層が含まれ、したがって、この論理ボリュームは複数層論理ボリュームになっている。最初のアグリゲーション層は、パーティションP1、L2、およびL3からのA1 51の作成と、パーティションP2、L4、およびL5からのA2 52の作成をもたらす。第2のアグリゲーション層は、A1とA2を単一のエンティティに組み合わせることによって、論理ボリューム、LV1 50を作る。これらのアグリゲーション層は、他の選択肢も容易に使用可能になるが、ソフトウェアRAIDまたはドライブ・リンクを表すことができる。既存のOS/2 LVMは、ボリュームの最初の区画の末尾のLVMデータ域に「偽」EBRを格納するはずである。しかし、複数レベルのアグリゲーションの場合、最下位のアグリゲータだけが、どのパーティションがボリューム内の最初のパーティションであるかを知っている。EBRに対する入出力要求を、インターセプトし、「偽」EBRにリダイレクトしなければならず、この検出およびリダイレクションが、最上位のアグリゲータで開始されなければならないので、「偽」EBRを見つける方法と、その後、「偽」EBRに入出力要求をリダイレクトする方法に関する疑問が生じる。本発明は、下で説明するブロードキャスト方法を使用することによって、この問題を解決する。

【0029】図7に、図5に示された論理ボリュームの例の、LVMデータ域のみかけの位置と、LVMデータ域の実際の位置を示す。最上位のアグリゲータ、LV1 50は、システム・ソフトウェアからは、LVMデータ域71をその末尾に配置されているように見える。しかし、LV1は、物理的に存在するのではなく、副アグリゲーションであるA2 52を含むディスク・パーティションの集合であるから、そのLVMデータ域は、まず、みかけ上、A2 52パーティションにマッピングされる。前に説明したように、A2パーティションも、それ自体のLVMデータ域を有し、これは、みかけ上はA2パーティション74の末尾に配置される。やはり、A2が物理的に存在しないので、A2のLVMデータ域およびマッピングされたLV1データ域の両方が、次の下位レベルの、最後のみかけ上のまたは実際のパーティションにマッピングされる。この場合、次の下位のみかけ上のパーティションは、実際のパーティションL5 59であり、したがって、LV1のLVMデータ域が、L5パーティションにマッピングされ、格納され(76)、その後、A2のLVMデータ域がマッピングされ、格納され(77)、最後に、実際のL5のLVMデータ域が、L5の末尾に配置される(78)。この例からわかるように、この方法では、最終的に実際のパーティションにマッピングされ、格納されるまで、すべての上位レベルLVMデータ域を下位レベルのみかけ上のパ

ーティションに連続的にマッピングすることによって、無限の数のアグリゲーション層が可能になる。

【0030】図10に、もう1つの例として、複数層論理ボリュームの機能スタック・モデルとツリー・モデルの間の相関を示す。この例では、機能スタックの最上位層が、暗号化機能159であり、その次が、ドライブ・リンク・アグリゲーション層160であり、その次が、RAIDなどの第2のアグリゲーション層161であり、その次が、不良ブロック再配置などのもう1つの機能層162であり、最後がバス・スルー層163である。図10の破線は、機能スタック層の層の、ツリー・モデルのノードおよびインターフェースへの対応を示し、この例では、ツリー・モデルに、最上位の論理ボリューム150、上位レベルのアグリゲータ151、第2レベル・アグリゲータ152および156の組、および複数のパーティション153、154、157、および158が含まれる。

【0031】論理ボリューム・サイズの変更—拡張および収縮

既存の複数層論理ボリュームのサイズを拡張または収縮する時に、特定の問題が生じる。たとえば、図5に示された論理ボリュームを、もう1つの副アグリゲーションA3および個別のパーティションP4を含むように拡張することができる。この論理ボリュームの拡張608の例を、図6に示す。拡張された状態の論理ボリュームLV1は、まだ下のアグリゲーション層によって作られる複数のアグリゲータからなるが、現在は、パーティションを直接に組み込まれてもいる。図示の例では、A1 51に、まだ3つの物理ディスク・パーティション、P1、L2、およびL3が含まれる。同様に、A2 52に、まだ3つのディスク・パーティション、P2、L4、およびL5が含まれ、追加されたA3 600に、2つのパーティションP3 604およびL6 608が含まれ、追加された個別のパーティションP4 606がある。

【0032】ボリュームのサイズおよび構成が変更されたので、LVMは、拡張されたアグリゲータのLVMデータ域内のデータを更新し、新しい区画および拡張処理の一部として作成されたアグリゲータのそれぞれにLVMデータ域を追加しなければならない。さらに、新しいパーティションまたはアグリゲータをその既存のアグリゲータに追加することによって拡張を実行するアグリゲータが、問題を有する。アグリゲータが拡張された後に、そのアグリゲータの既存のLVMデータ域は、もはやアグリゲータの末尾にない。したがって、このアグリゲータは、もはやパーティションのようには見えない。本発明は、アグリゲータが、ボリュームの機能スタック上でそれの上にあるすべての機能にとってパーティションであるように見えることを必要とするので、アグリゲータは、LVMデータ域を新たに拡張されたアグリゲ

トの末尾に移動しなければならない。

【0033】論理ボリュームが、図6に示されているように拡張される時に、LV1のLVMデータ域は、もやは論理ボリュームの最後のパーティション内に配置されておらず、したがって、LVMによって決定的に発見されることを可能にするために、LVMデータ域を再配置しなければならない。図8に移ると、LV1のLVMデータ域71は、元々はA2にマッピングされており、最終的に、LV1 50の拡張の前にはパーティションL5に格納されていた。副アグリゲーションであるA3 600およびパーティションP4 606が、LV1 50に追加されて、論理ボリュームが拡張される時に、LV1のLVMデータ域が、図示のように、論理ボリュームの新しい最後のパーティション、現在はP4に、再マッピングされ、格納される(80)。LVMデータ域の移動または再マッピングの実際の処理は、元のLVMデータ域の部分をコピーし、拡張された論理ボリュームの末尾で再作成することによって行われることが好ましい。これを行った後に、「偽」EBRを更新して、ファイルシステムおよびそれを使用するディスク・ユーティリティによって見られるボリュームの新しいサイズを反映することができる。この方法は、論理ボリュームの縮小について逆転することができる。

【0034】入出力要求処理のブロードキャスト方法
入出力要求を処理するブロードキャスト方法の処理を、図11に示す。EBRに対する入出力要求169が、複数レベルLVMによって検出された時に、その子の中で「偽」EBRが見つからないアグリゲータのそれぞれが、入出力要求を複製し、EBR入出力要求としてフラグを立て、入出力要求のその子のそれぞれに並列に発行する(127、172、173、174、175、および176)。この入出力要求の並列の複製および発行は、アグリゲーションの複数のレベルを下降することができる。すべての並列要求のうちで、1つだけが成功し、それ以外は破棄される。アグリゲータは、その子の中で「偽」EBRを見つけた時に、入出力要求を「偽」EBRにリダイレクトし、EBR入出力要求フラグをオフにする。入出力要求が、パス・スルー層に達した時に、EBR入出力要求フラグがセットされている場合には、パス・スルー層は、その入出力要求を破棄する。したがって、1つの入出力要求だけが、「偽」EBRに到達することに成功し、その途中で生成された複製入出力要求のすべてが、破棄される。この方法は、実施が単純であり、EBRに対する入出力要求は希なので、適度に効率的である。複製EBR入出力要求を並列に発行することに対する代替案は、それらを順次発行し、どれかが最初に成功した時点で停止することである。この場合、パス・スルー層は、EBR入出力フラグをセットされた入出力要求のすべてを、破棄するのではなくエラーにする。

【0035】要約

多数のオペレーティング・システムのLVMに有用な概念および方法論に適用可能な一般的な表現およびIBM社のOS/2オペレーティング・システムに適用可能な特定の表現の両方で、コンピュータ・システム用の複数層論理ボリューム・マネージャを実現する方法およびシステムを説明し、示した。

【0036】まとめとして、本発明の構成に関して以下の事項を開示する。

【0037】(1) コンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法であって、複数のコンピュータ可読媒体パーティションを設けるステップと、前記複数のコンピュータ可読媒体パーティションの複数のパーティション・アグリゲーションへの第1アグリゲーションを実行するステップと、機能強化された可用性およびコンピュータ・システムへのアクセス可能性のために、前記コンピュータ可読パーティションが、単一のコンピュータ可読エンティティにグループ化されるように、前記パーティション・アグリゲーションを複数層論理ボリュームにアグリゲートするステップとを含む方法。

(2) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、前記媒体パーティション間のドライブ・リンクを設けるステップを含む、上記(1)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(3) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、少なくとも1つのRAIDアレイを設けるステップを含む、上記(1)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(4) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記ステップが、前記媒体パーティション間のディスク・ミラーリングを設けるステップを含む、上記(1)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(5) 前記複数層論理ボリュームの編成を示す情報を含む論理ボリューム・データ域を設けるステップと、前記コンピュータ可読媒体パーティションの1つの中で拡張ブート・レコード内のリンクを設けるステップであって、前記リンクが、前記論理ボリューム内の前記論理ボリューム・データ域の位置を示すステップと、前記拡張ブート・レコードから、前記リンクを取り出すステップと、前記複数層論理ボリュームの前記編成を判定できるように、前記論理ボリューム・データ域から前記情報を取り出すステップとをさらに含む、上記(1)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(6) 前記論理ボリューム・データ域から前記情報を取り出す前記ステップが、さらに、少なくとも1つのアグリゲータに要求をブロードキャストすることを含む、上記(5)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(7) 前記論理ボリューム・データ域を前記論理ボリュームの最後のパーティションの末尾に格納するステップをさらに含む、上記(5)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理する方法。

(8) コンピュータ可読媒体の論理ボリュームを作成し、管理するシステムであって、少なくとも1つのコンピュータ可読媒体パーティションを有する少なくとも1つのコンピュータ可読媒体装置と、コンピュータ・ソフトウェアを実行することができ、少なくとも1つのコンピュータ可読媒体装置にインターフェースされる、コンピュータ・プロセッサと、前記コンピュータ可読媒体パーティションの複数層のアグリゲーションが、前記コンピュータ・プロセッサおよび前記コンピュータ・プロセッサによって実行されるソフトウェアによって、単一のエンティティとして論理的にアクセス可能になるように組み合わせられるように、複数層のコンピュータ可読媒体パーティション・アグリゲータを含む複数層論理ボリューム・マネージャとを含むシステム。

(9) 前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、コンピュータ可読媒体のRedundant Array of Independent Disks (「RAID」) アレイを含む、上記(8)に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

(10) 前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、ディスク・ミラーリング・サブシステムを含む、上記(8)に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

(11) 前記複数層論理ボリューム・マネージャ・アグリゲータが、さらに、ドライブ・リンク・サブシステムを含む、上記(8)に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

(12) 前記複数層論理ボリューム・マネージャが、機能強化されたIBM OS/2 LVMを含む、上記(8)に記載のコンピュータ可読媒体の論理ボリュームを作成し、管理するシステム。

(13) コンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラムであって、前記論理ボリュームが、複数のコンピュータ可読媒体パーティションを有し、前記コンピュータ・プログラムが、前記複数のコンピュータ可読媒体パーティションの複数のパーティション・アグリゲーションへの第1アグリゲーションを実行する、前記媒体内で実施されるコンピュータ可読プログラム・コード

手段を有するコンピュータ使用可能媒体と、機能強化された可用性およびコンピュータ・システムへのアクセス可能性のために、前記コンピュータ可読パーティションが、単一のコンピュータ可読エンティティにグループ化されるように、前記パーティション・アグリゲーションを複数層論理ボリュームにアグリゲートする、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体とを含むコンピュータ・プログラム。

(14) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、前記媒体パーティション間のドライブ・リンクを設けるコンピュータ可読プログラム・コード手段をさらに含む、上記(13)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

(15) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、少なくとも1つのRAIDアレイを設けるコンピュータ可読プログラム・コード手段をさらに含む、上記(13)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

(16) コンピュータ可読媒体パーティションの第1アグリゲーションを実行する前記コンピュータ可読プログラム・コード手段が、前記媒体パーティション間のディスク・ミラーリングを設けるコンピュータ可読プログラム・コード手段をさらに含む、上記(13)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

(17) 前記複数層論理ボリュームの編成を示す情報を含む論理ボリューム・データ域を設ける、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、前記コンピュータ可読媒体パーティションの1つの中で拡張ブート・レコード内のリンクを設けるコンピュータ可読プログラム・コード手段であって、前記リンクが、前記論理ボリューム内の前記論理ボリューム・データ域の位置を示す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、前記拡張ブート・レコードから、前記リンクを取り出す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体と、前記複数層論理ボリュームの前記編成を判定できるように、前記論理ボリューム・データ域から前記情報を取り出す、前記媒体内で実施されるコンピュータ可読プログラム・コード手段を有するコンピュータ使用可能媒体とをさらに含む、上記(13)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管

理するコンピュータ・プログラム。

(18) 前記論理ボリューム・データ域から前記情報を取り出す前記コンピュータ可読プログラム・コード手段が、さらに、少なくとも1つのアグリゲータに要求をブロードキャストするプログラム・コード手段を含む、上記(17)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

(19) 前記論理ボリューム・データ域を前記論理ボリュームの最後のパーティションの末尾に格納するコンピュータ可読プログラム・コード手段をさらに含む、上記(17)に記載のコンピュータ・システム用のコンピュータ可読媒体の論理ボリュームを作成し、管理するコンピュータ・プログラム。

【図面の簡単な説明】

【図1】パーソナル・コンピュータなどのコンピュータの基本的ハードウェア・アーキテクチャを示す図である。

【図2】そのようなコンピュータの基本的ソフトウェア・アーキテクチャを示す図である。

【図3】コンピュータ・ディスク表面のトラックへのフォーマットの性質を示す図である。

【図4】ディスク・セクタのブート・レコードおよびパーティションへの編成を示す図である。

【図5】論理ボリュームの偽拡張ブート・レコードの、グループ内の最初のセクタの末尾への配置を含む、論理ボリューム・マネージャによって使用される単一レベル・アグリゲーション方式を示す図である。

【図6】論理ボリュームの拡張を示す図である。

【図7】パーティションの論理ボリュームへの高水準ア

グリゲーション用のLVMデータ域の実際の物理記憶位置を示す図である。

【図8】論理ボリュームの拡張の後の最上位アグリゲータ用のLVMデータ域の再配置方法を示す図である。

【図9】LVMの機能スタック・モデルとLVMのツリー・モデルの間の周知の関係を示す図である。

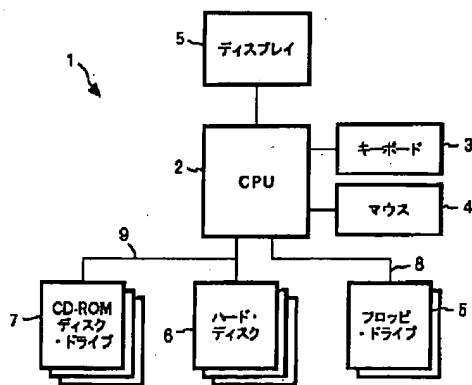
【図10】複数層LVMの機能スタック・モデルおよびツリー・モデルを示し、特にこの2つのモデルの対応する点を示す図である。

【図11】LVMデータ域を突きとめるために複数層LVMによって使用されるブロードキャスト方法をグラフィカルに示す図である。

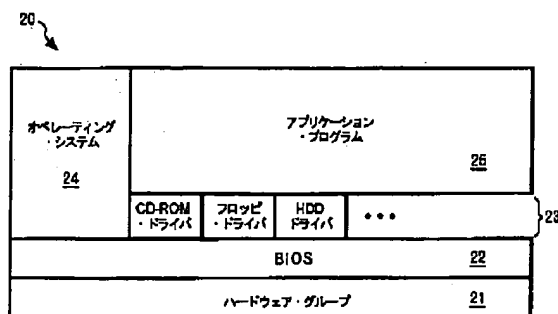
【符号の説明】

50 LV1
51 A1
52 A2
59 パーティションL5
150 論理ボリューム
151 アグリゲータ
152 第2レベル・アグリゲータ
153 パーティション
154 パーティション
156 第2レベル・アグリゲータ
157 パーティション
158 パーティション
159 暗号化機能
160 ドライブ・リンク・アグリゲーション層
161 第2のアグリゲーション層
162 機能層
163 パス・スルー層

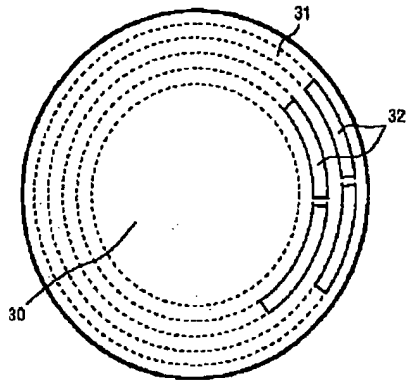
【図1】



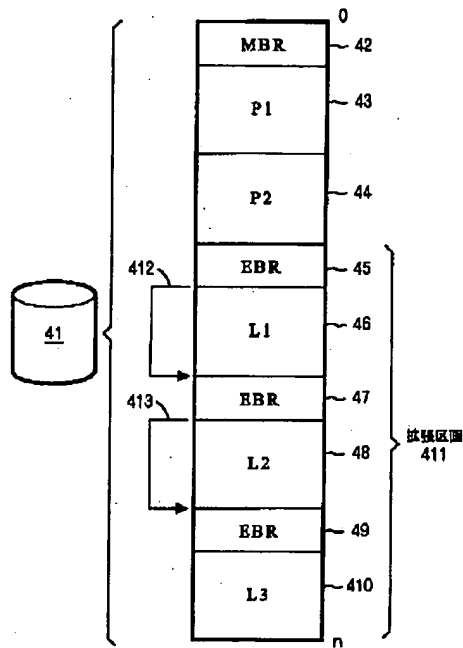
【図2】



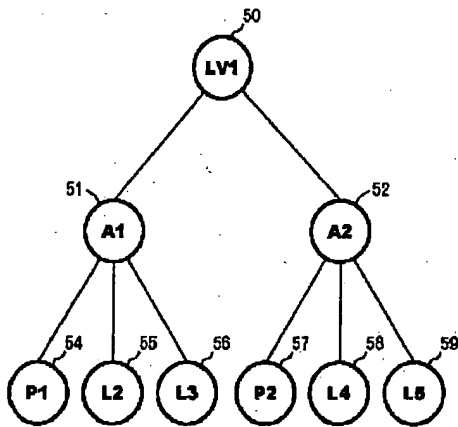
【図3】



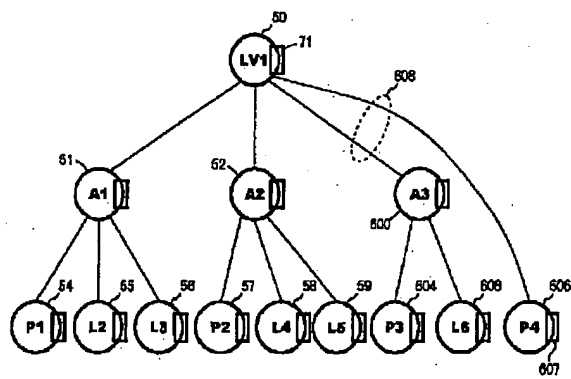
【図4】



【図5】

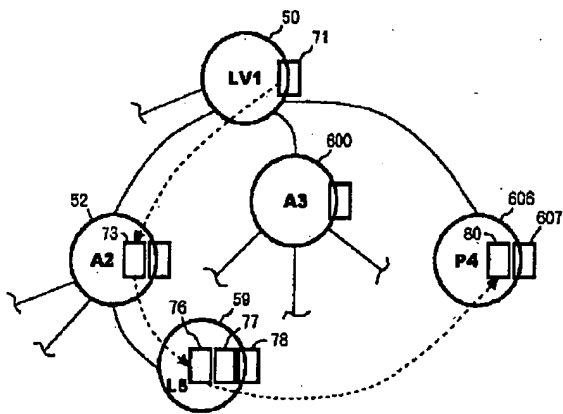
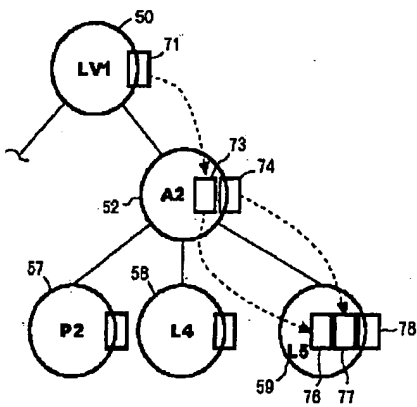


【図6】

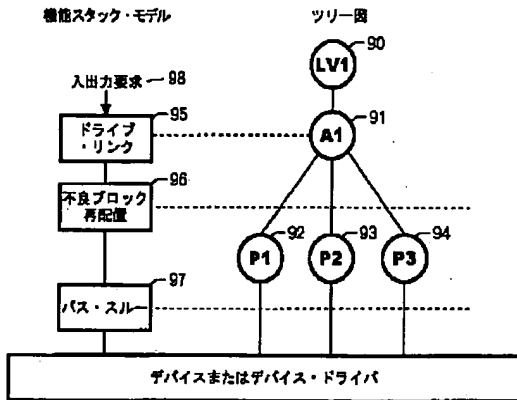


【図8】

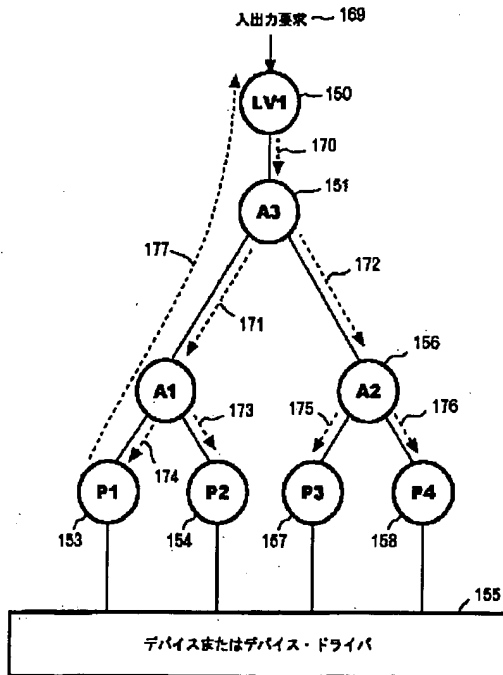
【図7】



【図9】



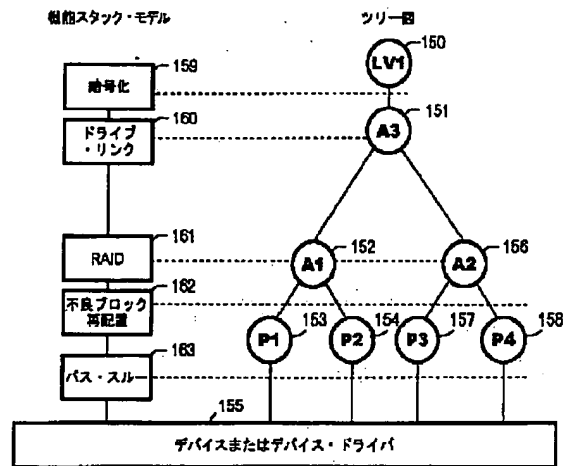
【図11】



フロントページの続き

(72)発明者 マーク・エイ・ペロクイン
 アメリカ合衆国78727 テキサス州オース
 チン イクエストリアン・コーヴ 13429
 (72)発明者 クォン・フォー・トラン
 アメリカ合衆国78727 テキサス州オース
 チン スコフィールド・ファームズ・ドラ
 イブ 13047

【図10】



(72)発明者 クリスティ・ネスビット・ウルマン
 アメリカ合衆国78727 テキサス州オース
 チン ピアス・ロード 2803

Fターム(参考) 5B065 ZA01 ZA16
 5B082 EA01
 5D110 AA13 AA15 BB21 DA01 DA11
 DB03 DC02 DE04